

Stopping Criteria for Iterative Decoding based on Mutual Information

Jinhong Wu
Mobile Solutions lab
Samsung Information Systems America
San Diego, CA 92121
Email: jinhongwu@ieee.org

Branimir R. Vojcic and Jia Sheng
ECE Department
The George Washington University
Washington, DC 20052
Email: vojic@gwu.edu, jiasheng@gwu.edu

Abstract—In this paper we investigate stopping criteria for iterative decoding from a mutual information perspective. We introduce new iteration stopping rules based on an approximation of the mutual information between encoded bits and decoder soft output. The first type stopping rule sets a threshold value directly on the approximated mutual information for terminating decoding. The threshold can be adjusted according to the expected bit error rate. The second one adopts a strategy similar to that of the well known cross-entropy stopping rule by applying a fixed threshold on the ratio of a simple metric obtained after each iteration over that of the first iteration. Compared with several well known stopping rules, the new methods achieve higher efficiency.

Index Terms—Iterative decoding, iteration stopping rule, mutual information

I. INTRODUCTION

Capacity approaching error correction coding schemes such as Turbo codes and LDPC codes are widely adopted in wireless standards, e.g., Turbo codes in 3GPP High Speed Packet Access (HSPA) and Long-Term Evolution (LTE) [1] [2], LDPC codes in WiMax [3] and Wi-Fi [4]. Iterative decoding is the practical solution for decoder implementation in modem chipsets. In order to achieve longer battery life and higher throughput, it is necessary to minimize chipset power consumption and processing delay. Iteration stopping rules serve such purposes by reducing the number of decoding iterations while maintaining the performance. In practice, cyclic redundancy check (CRC) is often employed for error detection which provides an easy solution for early stopping. However, not all systems have CRC at physical layer. For example, HSPA provides a 24-bit CRC for each transport block but no CRC for each code block within a transport block. A stopping criterion avoids decoding with the maximum number of iterations for every code block. Furthermore, while long CRC (e.g., 32-bit CRC) requires higher overhead, short CRC (e.g., 8-bit) results in weaker error detection. Iteration stopping without CRC is therefore of practical interest. In the following, we consider Turbo decoding, but the methodology applies as well to LDPC decoding [5].

Iteration stopping has been studied since early days of Turbo codes. The well known cross-entropy (CE) stopping rule [6] uses the relative information between the two constituent decoders' soft output as the criterion. Decoding is considered

as converged and stopped when this relative information is close to zero. Based on the same concept, two simplified variants of the CE rule were introduced in [7]: The first one, sign change ratio (SCR) rule, counts the number of sign changes in the extrinsic log likelihood ratios (LLRs) between two consecutive iterations. Decoding is terminated when the ratio is small enough. The second one, hard-decision-aided (HDA) rule, compares second decoder output hard decisions with those by the previous iteration. Decoding stops if all hard decisions remain the same. The overall performance by the simplified variants are close to those obtained by the original CE rule. Further variants or improvements include: sign difference ratio (SDR) [8] extends from SCR by comparing sign changes of each component decoder's a priori LLRs and extrinsic LLRs; improved hard-decision-aided (IHDA) rule [9] modifies HDA to compare hard decisions of two component decoders. Aside from savings in memory, the latter two variants perform soft/hard decision comparisons after every half iteration while previous methods perform comparisons after each complete iteration. While other stopping rules are also found in the literature, in e.g., [10] and references therein, it is interesting to notice that, above CE, SCR, SDR, HDA, and IHDA stopping rules all originate from the cross-entropy or relative information perspective of decoding convergence.

In this paper, we consider decoding convergence estimation from a mutual information perspective. In fact, mutual information analysis has been exemplified by the popular extrinsic information transfer (EXIT) chart [11]. In EXIT analysis, the mutual information between LLRs and the transmitted bit is calculated for system performance evaluation. While it is designed as an off-line tool, a similar and simplified process may be considered for online convergence estimation. In this paper, we show that instead of using true transmitted bits, a simple approximation of the mutual information by using decoder's hard decision can be used for efficient iteration stop.

The paper is organized as follows. The system model is described in Section II along with the approximated calculation of mutual information between encoded bit and LLR. The first iteration stopping rule is formulated in Section III. The second stopping rule is presented in Section IV. Simulations are shown in sub-sections III-B and IV-A. Finally, we conclude in Section V.

II. SYSTEM MODEL AND AN APPROXIMATE MUTUAL INFORMATION CALCULATION

We consider parallel concatenated convolutional code (PCCC) [12] where coded bits are mapped to symbols from a signal constellation and transmitted over a memoryless channel with additive white Gaussian noise (AWGN). At the receiver, assume perfect channel knowledge and optimal demodulation, iterative Log-MAP decoding [13] [6] is performed. The mutual information between each information bit $u \in \{1, -1\}$ and its associated LLR Λ can be derived as [11]

$$I = 1 - E_{\Lambda|\mu_\Lambda} \{\log_2[1 + \exp(-\Lambda)]\}. \quad (1)$$

where Λ denotes LLR in general, from which we may further specify a priori LLR, Λ_a , extrinsic LLR, Λ_e , and a posteriori LLR Λ_{app} by adding appropriate subscripts.

By applying the ergodicity assumption on LLR distribution, (1) is simplified as [14]

$$I \approx 1 - \frac{1}{N} \sum_{n=1}^N \log_2[1 + \exp(-u_n \Lambda(n))] \quad (2)$$

where $\Lambda(n)$ refers to the a priori LLR for bit u_n , $n = 1, 2, \dots, N$.

Convergence analysis based on above mutual information evolution has been utilized by the well-known EXIT chart [11] [14]. It takes the mutual information between the information bit and the associated a priori LLR, I_a , as the input, and generates the mutual information between the information bit and its extrinsic LLR, I_e , as the output. An EXIT chart is generated by plotting the output I_e values corresponding to a sequence of input $I_a \in [0, 1]$. A high mutual information indicates high reliabilities of LLRs, and vice versa.

Note that (2) requires knowledge of the information bits and therefore is used for off-line analysis. A blind mutual information calculation is presented in [15]. However, it requires more computations. For simple online analysis, we use the decoder hard decisions as estimates of the information bits, which results in an approximation of the desired mutual information I . Clearly, its accuracy depends on the reliability of the hard decisions. However, since output LLRs provides probabilistic measure about information bits, this approximation can approach the maximum value only if LLRs are large enough, which in turn indicates the decoding convergence [16]. Therefore, for the purpose of iteration stopping, a high threshold value on the approximated mutual information can be effective. On the other hand, we are interested in knowing the reliability of the overall decoder output, rather than the extrinsic information alone. Therefore we may consider the mutual information generated by a posteriori LLR, Λ_{app} . In fact, using Λ_{app} instead of Λ_e allows for earlier identification of decoding convergence.

Applying the hard decision by $\hat{u} = \text{sign}(\Lambda_{app})$, we approximate (2) of a posteriori LLR, I_{app} , by

$$\begin{aligned} \hat{I}_{app} &\approx \frac{1}{N} \sum_{n=1}^N (1 - \log_2(1 + \exp(-\Lambda_{app}(n) \cdot \text{sign}(\Lambda_{app}(n)))) \\ &= 1 - \frac{1}{N \ln(2)} \sum_{n=1}^N \ln(1 + e^{-|\Lambda_{app}(n)|}) \\ &\approx 1 - \frac{1.44}{N} \sum_{n=1}^N \ln(1 + e^{-|\Lambda_{app}(n)|}) \\ &= 1 - \epsilon \end{aligned} \quad (3)$$

where we define $\epsilon = \frac{1.44}{N} \sum_{n=1}^N \ln(1 + e^{-|\Lambda_{app}(n)|})$.

Furthermore, adopting the common practice in the log-MAP algorithm implementations [6], the calculation of $\ln(1 + \exp(-|A|))$ can easily be handled by a look up table function, $LUT(|A|)$. Therefore, (4) is simplified as

$$\hat{I}_{app} \approx 1 - \frac{1.44}{N} \sum_{n=1}^N LUT(|\Lambda_{app}(n)|). \quad (4)$$

Note that due to the averaging over all bit decisions, this approximation becomes more accurate as the iterative decoding converges to the reliable decision. When the decoding delivers reliable results, $\Lambda_{app}(n)$ values are generally high. Consequently, $\hat{I}_{app} \rightarrow 1$ by (4). In contrast, erroneous decoding will result in a lower \hat{I}_{app} value due to smaller magnitudes of LLRs. For illustration purpose, three typical examples are shown in Fig.1 - 2: They are based on decoding a (7, 5) code using random interleavers of length 900. Coded data are binary phase shift keying (BPSK) modulated and transmitted over AWGN channel. Fig. 1 is obtained from unsuccessful decoding of a packet at $E_b/N_0 = 1\text{dB}$. The inner two curves in Fig. 2 are from successful decoding of another packet also at $E_b/N_0 = 1\text{dB}$, while the outer two curves in Fig. 2 are from successful decoding of a packet at $E_b/N_0 = 3\text{dB}$. The evolution of \hat{I}_{app} during the iterations is plotted: The vertical axis, denoted by \hat{I}_{app}^1 , refers to the mutual information produced by the 1st constituent decoder. The horizontal axis, denoted by \hat{I}_{app}^2 , refers to that produced by the 2nd constituent decoder. By sequential exchange of extrinsic information, each decoder accepts the mutual information produced by the other decoder as the input, and generates its own mutual information as the output. The two curves in each plot indicate how each decoder's output mutual information evolves according to the input mutual information it receives from the other decoder.

III. ϵ THRESHOLD AND TYPE-I STOPPING RULE

Based on the characteristics shown above, we propose a mutual information aided (MIA) iteration stopping rule: decoding stops when \hat{I}_{app}^2 is very close to 1, i.e., ϵ is small enough.

¹ We denote this stopping rule as type-I mutual information

¹It is reasonable to apply the threshold on \hat{I}_{app} also for 1st decoder, which will result in decoding stops 0.5 iterations earlier for some cases.

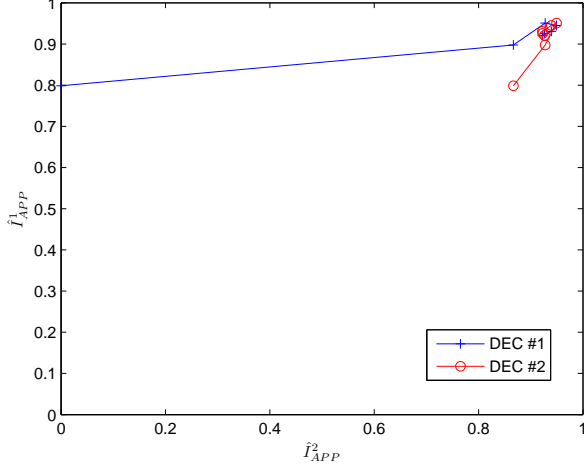


Fig. 1. Evolution of the approximated mutual information: Decoding convergence not reached ($E_b/N_0 = 1\text{dB}$).

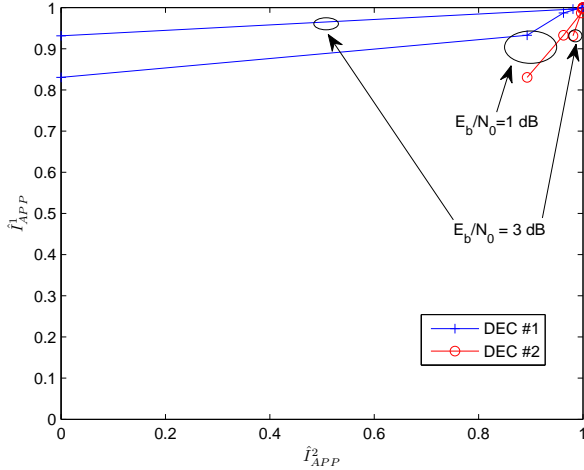


Fig. 2. Evolution of the approximated mutual information: Decoding convergence reached at low SNR ($E_b/N_0 = 1\text{dB}$) and high SNR ($E_b/N_0 = 3\text{dB}$).

aided (MIA-I) rule. For simplicity, a fixed threshold value on ϵ (e.g., 10^{-5}) may be applied. For flexible iteration stopping, more specifics about ϵ are considered as follows.

A. Estimation of Bit error rate and ϵ threshold

According to the definition of LLR, it is straightforward to derive the conditional bit error probability given the a posteriori LLR Λ as

$$P_{e|\Lambda} = \frac{1}{1 + e^{|\Lambda|}}. \quad (5)$$

For large $|\Lambda|$ it then follows that

$$P_{e|\Lambda} = \frac{1}{1 + e^{|\Lambda|}} \approx \frac{1}{e^{|\Lambda|}} \approx \ln(1 + e^{-|\Lambda|}). \quad (6)$$

For reasonably large $|\Lambda|$ (e.g., for $|\Lambda| \geq 2$), we can verify that the difference from approximation is negligible. Therefore ϵ in

(4) could also be used for a rough estimation of decoding bit error rate (BER). Note that LLRs produced by early iterations are more accurate but less so in later iterations due to increased correlations with soft input to the decoder(s) [11]. For this reason, ϵ approximates BER better in early iterations but not as well in later iterations (as it usually approaches zero with a large number of iterations). For effective iteration stopping, the threshold ϵ may be selected to be around the value of expected BER. Consider a fixed threshold of ϵ , it is intuitively clear that a lower threshold ensures lower BER but may require more iterations. This provides the following two options of selecting a threshold of ϵ .

Option A: When decoding throughput is of higher priority but BER is less important, a relatively high threshold of ϵ can be applied. This may be appropriate for certain real time applications such as voice or video transmissions with certain quality of service (QoS) requirement. In some practical scenarios BER of 10^{-3} to 10^{-4} can be acceptable. In those scenarios, a fixed threshold on ϵ may be chosen from, e.g., 10^{-2} to 10^{-5} , for the desired BER. For notation purpose we denote MIA-I rule with fixed ϵ threshold by MIA-I-A.

Option B: When decoding reliability is of higher priority, an estimation of achievable BER can be used for ϵ threshold. For known channel and coding scheme, BER can be measured in advance. If such a BER value is unknown a priori, a sufficiently low ϵ may be used initially, and then adjusted accordingly once BER can be measured or estimated. This may apply to channels with slow changes. We denote MIA-I rule with such adaptive ϵ thresholds by MIA-I-B.

B. Simulations

The effects of different thresholds are illustrated by simulations. The PCCC encoding scheme uses two identical (7,5) component encoders, with random interleaving of size 900. Puncturing of even (odd) indexed parity bits by first (second) encoder is applied to generate the rate 1/2 code. Coded data are modulated with BPSK and transmitted over AWGN channel.

For MIA-I-A, we applied $\epsilon = 10^{-2}$, $\epsilon = 10^{-3}$, and $\epsilon = 10^{-4}$ for different trade-offs between BER and number of iterations. For MIA-I-B, we adaptively set ϵ according to Table I to match the achievable BER. We compare these MIA stopping rules with the CE rule as well as the HDA rule^{2,3}. The threshold for the CE criterion is 10^{-4} . All stopping rules are then compared with decoding using 6 iterations.

We plot BER curves as well as the average number of iterations in Fig. 3 and Fig. 4. As a reference, BER as well as the average numbers of iterations assuming a ‘genie’ error detector are also plotted. Its number of iterations refers to either the lowest number of iterations for error free decoding, or the maximum number of 6 iterations if errors always exist.

²Other stopping methods, e.g., SCR, SDR or IHDA, etc, are similar to or slightly worse than CE or HDA in performance and/or average number of iterations as reported in [7], [8] and [9].

³Also note that the proposed MIA rules can also compare \hat{I}_{app} or ϵ ratios after each half iteration with trivial modification, which could further reduce the average iteration number.

TABLE I
THRESHOLD OF ϵ FOR DIFFERENT SNR

(7,5) code with interleaver size 900, AWGN channel

E_b/N_0	1	2	3	4
ϵ	10^{-2}	10^{-4}	10^{-5}	10^{-6}

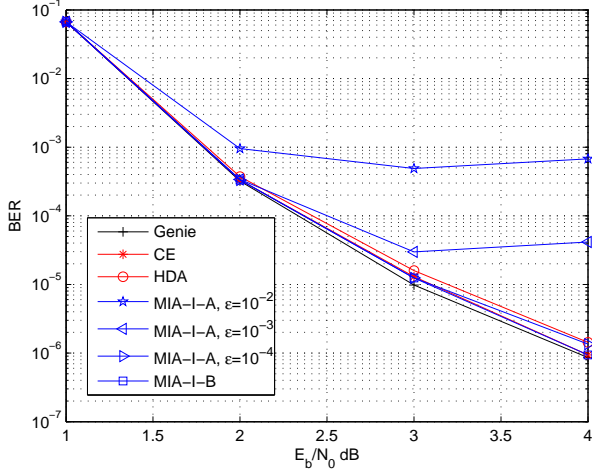


Fig. 3. Performance by different ϵ threshold: rate 1/2 turbo code with memory length 2, interleaver size 900, over AWGN channel.

IV. ϵ RATIO THRESHOLD AND TYPE-II STOPPING RULE

To avoid estimation of BER for determination of \hat{I}_{app} threshold, we further consider the metric ϵ . As shown by Fig. 2, decoding convergence accompanies the minimization of ϵ . Interestingly, in average $\epsilon(iter)$ shows a similar pattern as that of the CE criterion [6] that we can leverage.

To illustrate we briefly review the CE rule below. Denote

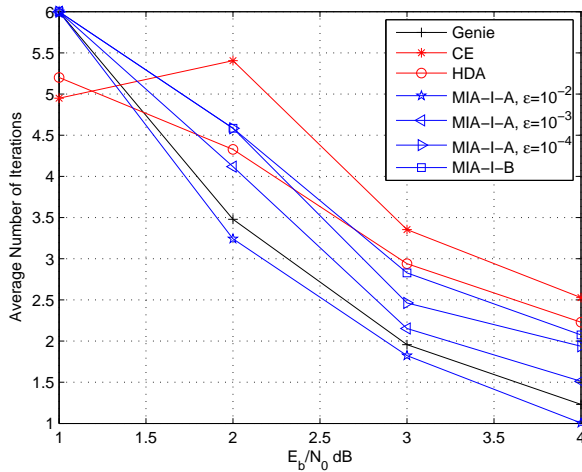


Fig. 4. Average number of iterations by different ϵ threshold: rate 1/2 turbo code with memory length 2, interleaver size 900, over AWGN channel.

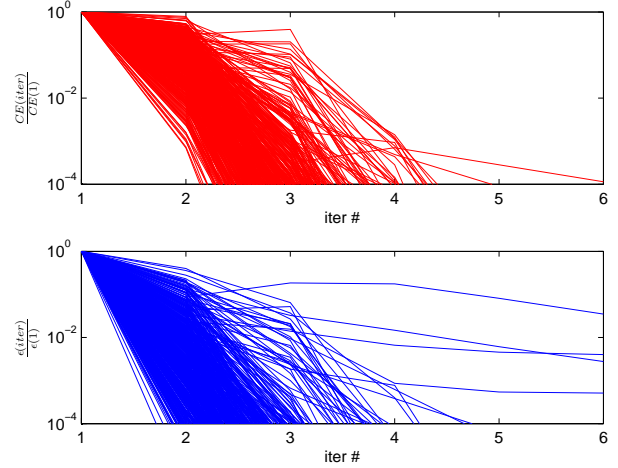


Fig. 5. cross-entropy ratio and ϵ ratio over iterations: rate 1/2 turbo code with memory length 2, interleaver size 900, over AWGN channel.

iteration number as $iter$ and consider extrinsic and a posteriori LLRs output from second component decoder, CE is expressed as [6]

$$CE(iter) = \frac{1}{N} \sum_{n=1}^N \frac{|\Delta \Lambda_e^{iter}(n)|^2}{e^{\Lambda_{app}^{iter}(n)}} \quad (7)$$

where $\Delta \Lambda_e^{iter} = \Lambda_e^{iter} - \Lambda_e^{iter-1}$.

The effectiveness of CE rule comes from a fact that the ratio of $\frac{CE(iter)}{CE(1)}$ typically shows an accelerated decreasing as decoding converges. Decoding stops if $\frac{CE(iter)}{CE(1)} < 10^{-3}$ or $\frac{CE(iter)}{CE(1)} < 10^{-4}$. Usually choosing 10^{-3} saves slightly in iteration numbers but more often result in an early error floor, while 10^{-4} usually maintain performance better at the cost of slightly higher number of iterations.

To show the pattern of $\frac{\epsilon(iter)}{\epsilon(1)}$ in comparison to $\frac{CE(iter)}{CE(1)}$, we plot numerically such two ratios over iterations in Fig. 5. The figures are randomly generated by decoding 500 coded packets at $E_b/N_0 = 3$ dB and measuring those two ratios by each packet after each iteration. In both figures one curve is plotted for each decoded packet. In average, $\frac{\epsilon(iter)}{\epsilon(1)}$ drops faster. This implies the possibility of earlier stopping compared with the CE rule.

Based on the observation, we express the second stopping criterion as $\frac{\epsilon(iter)}{\epsilon(1)} < 10^{-3}$. We denote this stopping rule as type-II mutual information aided (MIA-II) rule. We further note that computation-wise, $\epsilon(iter)$ requires N look-up table search, $N - 1$ additions, and one multiplication, while $CE(iter)$ requires N times look-up table search (for the exponential function), $2N - 1$ additions, and $2N + 1$ multiplications.

A. Simulations

In this section we compare MIA-II and MIA-I-B with the CE rule as well as the HDA rule. Simulations are based on PCCC, with the (7, 5) component code with random

TABLE II
 ϵ THRESHOLD VALUES FOR DIFFERENT SNR

(7,5) code with interleaver size 2048, AWGN channel

E_b/N_0	1	2	3	4	5
ϵ	10^{-1}	10^{-3}	10^{-5}	10^{-6}	10^{-7}

(7,5) code with interleaver size 2048, Rayleigh fading channel

E_b/N_0	3	4	5	6	7
ϵ	2×10^{-2}	2×10^{-4}	2×10^{-5}	5×10^{-6}	10^{-6}

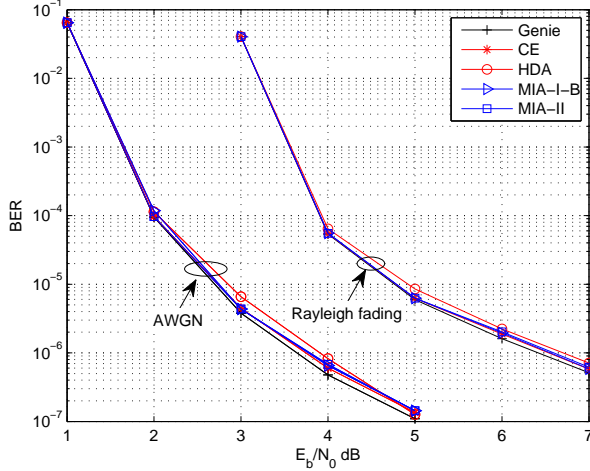


Fig. 6. BER: rate 1/2 turbo code with memory length 2, interleaver size 2048, over AWGN channel and fast Rayleigh fading channel.

interleavers of size 2048. We compare different stopping rules for transmissions by BPSK over AWGN channel and fast Rayleigh fading channel.

For MIA-I-B stopping rule, the threshold value is chosen as in Table II. The BER performance comparison is shown in Fig. 6. The average numbers of iterations by each rule are shown in Fig. 7. Compared with the CE rule or the HDA rule, the proposed MIA rules usually stops earlier.

V. CONCLUSIONS

Using turbo code as example, we provided an approximate calculation of the mutual information between encoded bit and decoder's a posteriori LLR. The changing pattern of such metric effectively indicates decoding convergence after iterations. Two types of iteration stopping rules are proposed. The first type includes two options: MIA-I-A using a fixed threshold provides flexible trade-off between performance and complexity; MIA-I-B ensures the performance by an estimation of the achievable BER. The second type, MIA-II, adopts a more universal threshold with a strategy similar to that of the CE rule, but requires less computation. Simulations show that compared with CE and HDA stopping rules, both MIA-I-B and MIA-II achieve higher efficiency while maintaining achievable performance by a maximum number of iterations.

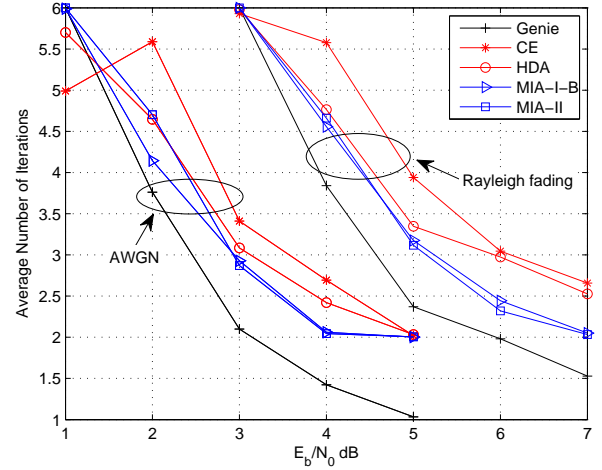


Fig. 7. Average number of iterations: rate 1/2 turbo code with memory length 2, interleaver size 2048, over AWGN channel and fast Rayleigh fading channel.

REFERENCES

- [1] 3GPP, "High Speed Downlink Packet Access: Physical Layer Aspects (Release 5)," TR 25.858, Tech. Rep., 2002.
- [2] —, "Evolved universal terrestrial radio access (E-UTRA); multiplexing and channel coding (Release 10)," TS 36.212, Tech. Rep., 2011.
- [3] IEEE, "Air interface for fixed and mobile broadband wireless access systems," P802.16e/D12, Tech. Rep., Oct. 2005.
- [4] —, "Wireless lan medium access control and physical layer specifications: Enhancements for higher throughput," P802.16n/D1.0, Tech. Rep., Mar. 2006.
- [5] J. Li, X. You, and J. Li, "Early stopping for ldpc decoding: Convergence of mean magnitude (cmm)," *IEEE Communications Letters*, vol. 10, pp. 667–669, Sep. 2006.
- [6] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inform. Theory*, vol. 42, no. 2, pp. 429–445, Feb. 1996.
- [7] R. Shao, S. Lin, and M. Fossorier, "Two simple stopping criteria for turbo decoding," *IEEE Trans. Commun.*, vol. 47, pp. 1117–1120, Aug. 1999.
- [8] Y. Wu, B. Woerner, and J. Ebel, "A simple stopping criterion for turbo decoding," *IEEE Communications Letters*, vol. 4, pp. 258–260, Aug. 2000.
- [9] T. Ngatched and R. Takawira, "Simple stopping criterion for turbo decoding," *Electronics Letters*, vol. 37, pp. 1350–1351, Oct. 2001.
- [10] E. Boutillon, C. Douillard, and G. Montorsi, "Iterative decoding of concatenated convolutional codes: Implementation issues," *Proceedings of the IEEE*, vol. 95, pp. 1201–1227, Jun. 2007.
- [11] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Commun.*, vol. 49, pp. 1727–1737, Oct. 2001.
- [12] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo codes," in *Proc. IEEE Intl. Conf. on Com.*, Geneva, Switzerland, May 1993, pp. 1064–1070.
- [13] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. 22, pp. 284–287, Mar. 1974.
- [14] J. Hagenauer, "The exit chart - introduction to extrinsic information transfer in iterative processing," in *European Signal Processing Conference*, Vienna, Austria, Sep. 2004, pp. 1541–1548.
- [15] I. Land, P. A. Hoeher, and S. Gligorevic, "Computation of symbol-wise mutual information in transmission systems with logapp decoders and application to exit charts," in *Proc. of Intl. ITG Conf. on Source and Channel Coding*, Erlangen, Germany, Jan. 2004, pp. 195–202.
- [16] A. Reid, T. Gulliver, and D. Taylor, "Convergence and errors in turbo-decoding," *IEEE Trans. Commun.*, vol. 49, pp. 2045–2051, Dec. 2001.